Appendix A. File Formats

Grid File Format

The grid files **grid.in** and **x.save** conform to the PLOT3D single- or multiple-grid format, and may or may not contain IBLANK. The single-grid file format (with IBLANK) can be read using the following Fortran statements:

```
      READ(1) JD,KD,LD
      READ(1) (((X(J,K,L),J=1,JD),K=1,KD),L=1,LD),
     &        (((Y(J,K,L),J=1,JD),K=1,KD),L=1,LD),
     &        (((Z(J,K,L),J=1,JD),K=1,KD),L=1,LD),
     &        (((IBLANK(J,K,L),J=1,JD),K=1,KD),L=1,LD)
```

The multiple-grid file format (again with IBLANK) is

```
      READ(1) NGRID
      READ(1) (JD(IG),KD(IG),LD(IG),IG=1,NGRID)
      DO IG = 1,NGRID
        READ(1) (((X(J,K,L),J=1,JD(IG)),K=1,KD(IG)),L=1,LD(IG)),
     &          (((Y(J,K,L),J=1,JD(IG)),K=1,KD(IG)),L=1,LD(IG)),
     &          (((Z(J,K,L),J=1,JD(IG)),K=1,KD(IG)),L=1,LD(IG)),
     &          (((IBLANK(J,K,L),J=1,JD(IG)),K=1,KD(IG)),L=1,LD(IG))
      ENDDO
```

The OVERFLOW code does not use the IBLANK contained in **grid.in** or **x.save**. The IBLANK information contained in **grid.in** or **x.save** is solely for post-processing.

Q File Format

The format of the flow information written by OVERFLOW 2 basically conforms to the PLOT3D Q file format. This is used for files **q.save**, **q.bomb**, and **q.restart**. The Q file will be in single- or multiple-grid format depending on the number of grids in **grid.in**. There are some differences between PLOT3D and OVERFLOW Q file formats, however, including additional flow information in the Q header record and additional Q variables such as γ and possible species densities and turbulence field variables.

The OVERFLOW 2 single-grid Q file format can be read using the following Fortran statements:

```
      READ(1) JD,KD,LD,NQ,NQC
      READ(1) REFMACH,ALPHA,REY,TIME,GAMINF,BETA,TINF,
     &        IGAM,HTINF,HT1,HT2,(RGAS(I),I=1,MAX(2,NQC)),
     &        FSMACH,TVREF,DTVREF
      READ(1) ((((Q(J,K,L,N),J=1,JD),K=1,KD),L=1,LD),N=1,NQ)
```

where NQ is the total number of Q field variables. NQ is composed of the conserved variables and the ratio of specific heats (ρ^*, ρ^*u^*, ρ^*v^*, ρ^*w^*, $\rho^*e_0^*$, γ), plus NQC species densities ρ^*c_i, plus the turbulence field quantities (NQT=1 for one equations models, NQT=2 for two equation models). Thus

$$\text{NQ} = 6+\text{NQC}+\text{NQT} \tag{A1}$$

JD, KD, and LD are the grid dimensions in J, K, and L. The second record includes the Q header information, including the reference Mach number $M_{ref}=V_{ref}/c_\infty$ (**REFMACH**), angle-of-attack α (**ALPHA**), Reynolds number Re (**REY**), and **TIME**, which OVERFLOW currently uses to store the iteration number. These four numbers are the standard PLOT3D Q header information. The OVERFLOW header includes additional information to support the variable gamma, force/moment, and species convection options, including free-stream gamma γ_∞ (**GAMINF**), side-slip-angle β (**BETA**), free-stream temperature T_∞ (**TINF**, in degrees Rankine), variable gamma option **IGAM**, free-stream stagnation enthalpy $h_0{}^*{}_\infty$ (**HTINF**), stagnation enthalpy ratios **HT1** and **HT2**, and species gas constants $R_i{}^*$ (**RGAS**). Also included are the free-stream Mach number $M_\infty=V_\infty/c_\infty$ (**FSMACH**), the simulation time (**TVREF**) and time step (**DTVREF**), both based on V_{ref}.

The multiple-grid Q file format is

```
 READ(1) NGRID
 READ(1) (JD(IG),KD(IG),LD(IG),IG=1,NGRID),NQ,NQC
 DO IG = 1,NGRID
   READ(1) REFMACH,ALPHA,REY,TIME,GAMINF,BETA,TINF,
&           IGAM,HTINF,HT1,HT2,(RGAS(I),I=1,MAX(2,NQC)),
&           FSMACH,TVREF,DTVREF
   READ(1) ((((Q(J,K,L,N),J=1,JD(IG)),K=1,KD(IG)),L=1,LD(IG)),N=1,NQ)
 ENDDO
```

The Q restart file also includes the Q values at the ISTEP-1 time step when 2nd-order time is used in conjunction with dual-time or Newton sub-iterations (**FSONWT**>1.0 and **NITNWT**>1). The multiple-grid Q file format is

```
 READ(1) NGRID
 READ(1) (JD(IG),KD(IG),LD(IG),IG=1,NGRID),NQ,NQC
 DO IG = 1,NGRID
   READ(1) REFMACH,ALPHA,REY,TIME,GAMINF,BETA,TINF,
&           IGAM,HTINF,HT1,HT2,(RGAS(I),I=1,MAX(2,NQC)),
&           FSMACH,TVREF,DTVREF
   READ(1) ((((Q(J,K,L,N),J=1,JD(IG)),K=1,KD(IG)),L=1,LD(IG)),N=1,NQ)
 ENDDO
 DO IG = 1,NGRID
   READ(1) REFMACH,ALPHA,REY,TIME,GAMINF,BETA,TINF,
&           IGAM,HTINF,HT1,HT2,(RGAS(I),I=1,MAX(2,NQC)),
&           FSMACH,TVREF,DTVREF
   READ(1) ((((QN(J,K,L,N),J=1,JD(IG)),K=1,KD(IG)),L=1,LD(IG)),N=1,NQ)
 ENDDO
```

Here Q contains the conserved variables at the current time level and QN contains the conserved variables at the previous time level.

The file **q.bomb** is written only in case a negative density or pressure is detected, if the specific heat ratio γ falls below one, or if the flow solver residual norm exceeds a limit value. Thus at the end of a run so terminated, **q.bomb** represents the solution at the time of the error, and **q.save**, if present, represents the most recent intermediate solution (saved based on the input parameter **NSAVE**).

Q-Average File Format

The **q.avg** file is generated if the NAMELIST input **ISTART_QAVG** is greater than 0. The file uses the modified Q file format described above. One record is appended to the file, with the number of steps that statistics were collected over. The output Q variables are the time averaged density, momentum, total energy, and ratio of specific heats (Q variables 1-6) and the perturbation of density, velocity, and pressure (ρ'^2, u'^2, v'^2, w'^2, p'^2) (Q variables 7-11). The **q.avg** is not written if the input variables **DYNMCS**=.TRUE. or **NADAPT**$\neq$0 (moving or adapting grid systems). The **q.avg** file is overwritten at restarts.

Chimera INTOUT or XINTOUT Interpolation File Formats

The interpolation file format used by PEGASUS 5 is called **XINTOUT**, and provides the interpolation and blanking information required by the flow solver to connect multiple grids. This format must be used for multiple-grid runs in OVERFLOW-mode. The format is

```
 DO IG = 1,NGRID
    READ(2) IBPNTS(IG),IIPNTS(IG),IIEPTR(IG),IISPTR(IG)
    READ(2) (JI(I),I=1,IIPNTS(IG)),(KI(I),I=1,IIPNTS(IG)),
&        (LI(I),I=1,IIPNTS(IG)),(DJ(I),I=1,IIPNTS(IG)),
&        (DK(I),I=1,IIPNTS(IG)),(DL(I),I=1,IIPNTS(IG))
```

```
   READ(2) (JB(I),I=1,IBPNTS(IG)),(KB(I),I=1,IBPNTS(IG)),
&         (LB(I),I=1,IBPNTS(IG)),(IBC(I),I=1,IBPNTS(IG))
   READ(2) (((IBLANK(J,K,L),J=1,JD),K=1,KD),L=1,LD)
 ENDDO
```

IBLANK here has a value of zero for hole or Chimera boundary points and one for all other points. IBPNTS is the number of Chimera boundary points in this grid, and IIPNTS is the number of interpolation points (or stencils) in this grid, used by Chimera boundary points in any grid. IISPTR and IIEPTR give start and end pointers for interpolated data from this grid into the global QBC array (Q data Chimera boundary array). (JI,KI,LI) gives the donor cell for interpolation data with (DJ,DK,DL) as linear weights for the Q variables between JI and JI+1, etc. The result of these interpolations gets stored into QBC elements IISPTR through IIEPTR. (JB,KB,LB) is the boundary point which receives interpolated data from QBC element IBC.

Thus, some sanity checks on this information include

$$IIPNTS = IIEPTR - IISPTR + 1 \tag{A2}$$

and

$$\sum_{IG} IBPNTS(IG) = \sum_{IG} IIPNTS(IG) \tag{A3}$$

In OVERFLOW-D-mode, the internal software DCF is used to generate Chimera interpolation information. This information is saved in the **INTOUT** file, which has a somewhat different format from **XINTOUT**, though the information is the same.

```
READ(2) IBPNTS(1:NGRID)
DO IG = 1,NGRID
   READ(2) (JB(I),I=1,IBPNTS(IG)),(KB(I),I=1,IBPNTS(IG)),
&         (LB(I),I=1,IBPNTS(IG)),(ID(I),I=1,IBPNTS(IG)),
&         (JI(I),I=1,IBPNTS(IG)),(KI(I),I=1,IBPNTS(IG)),
&         (LI(I),I=1,IBPNTS(IG)),(DJ(I),I=1,IBPNTS(IG)),
&         (DK(I),I=1,IBPNTS(IG)),(DL(I),I=1,IBPNTS(IG))
   READ(2) (((IBLANK(J,K,L),J=1,JD),K=1,KD),L=1,LD)
 ENDDO
```

Here all information relating to an interpolation stencil is given relative to the Chimera boundary point list: (JB,KB,LB) is the list of boundary points, ID is the list of donor grid number, (JI,KI,LI) are the donor cells, and (DJ,DK,DL) the corresponding offsets within the cells. There is no implied indexing into a global QBC array. IBLANK here has a value of zero for hole points, -1 for Chimera boundary points, and one for all other points.

DCF xrays.in File

The **xrays.in** file is required by DCF when running in the OVERFLOW-D mode. This file is used to cut holes and can be generated in the **overgrid** utility (see Chapter 4). The format is given here, and is also documented in Chimera Grid Tools.

```
C       Number of X-rays.
        READ(1) NXRAYS
C       Number of points in J,K, Body ID.
        READ(1) (JX(N),KX(N),IXBDY(N),N=1,NXRAYS)
C       (x,y,z) bounding boxes.
        DO N=1,NXRAYS
          READ(1) XBOXES(1:6,N)
        ENDDO
        DO N=1,NXRAYS
C         Total number of pierce-points for this X-ray.  If NXTOT<0, this X-ray is a duplicate of N=-NXTOT.
```

```
           READ(1) NXTOT(N)
           IF (NXTOT(N).GT.0) THEN
C             Start/end indices into XRAY for each (j,k) ray.
              READ(1) I_PTR(1:JX(N),1:KX(N),1:2)
C             List of pierce-point z-values (always in in/out pairs).
              READ(1) XRAY(1:NXTOT(N))
           ENDIF
         ENDDO
```

Residual and Other History File Formats

Note that these files are concatenated into **basename.resid**, etc., by the **overrun** and **overrunmpi** scripts, to provide a complete history of a simulation, including restarts.

The text file **resid.out** contains the flow solver (i.e., five primary variables) residual history. Columns in the file list grid#, step#, right-hand side l_2- and l_∞-norms, (j,k,l) location of the l_∞-norm, and ΔQ l_2- and l_∞-norms, and (j,k,l) location of the l_∞-norm. The final column is incremental wall clock time. See the OVERFLOW script **overrun** and related tools for convenient ways to process residual and other files into formats plottable by **xmgrace** or some similar line plot package. **overrun** is in the **tools/run** subdirectory.

Files **turb.out** and **species.out** list the (1- or 2-equation) turbulence model and species continuity equation residual histories, respectively, and include the same parameters as **resid.out**.

Minimum density, pressure, and gamma for each grid are tracked in file **rpmin.out**, along with a count of the number of reverse flow (u<0) and supersonic points. These values may aid in monitoring convergence history, and are included for diagnostic purposes. Columns in the file are grid#, step#, $\rho^*_{\rm min}$, p^*_{min}, γ_{min}, #reverse points, #supersonic points, and maximum turbulent eddy viscosity.

As implemented, residual history information in **resid.out**, **turb.out**, and **species.out** is recorded for each Newton sub-iteration, to allow inspection of the convergence of the sub-iteration process. History information in files **rpmin.out** and **fomoco.out** is only recorded for the last sub-iteration of each step.

BC#201 Output File Format

Boundary condition 201 (**IBTYP**=201) may be used to extract flow quantity information for multiple iterations of the flow solver. (**BCPAR1** can be used to specify the iteration number to start saving flow information, and **BCPAR2** can specify the iteration increment for saving. Information will be saved when *mod*(step#,BCPAR2)=0.) The output for any grid may be a point, line, surface, or volume subset of the grid. The output is written to a Fortran unformatted output file. The file may be read using the following

```
         DO N=1,ISTEPS
           READ(1) IGRID,ISTEP,NJ,NK,NL,NQ,NQC,TVREF,DTVREF,
   &        X(1:NJ,1:NK,1:NL),Y(1:NJ,1:NK,1:NL),Z(1:NJ,1:NK,1:NL),
   &        Q(1:NJ,1:NK,1:NL,1:NQ),IBLANK(1:NJ,1:NK,1:NL)
         ENDDO
```

The output file will be written to the file name specified in the **BCFILE** NAMELIST variable. If no name is specified, the file will be written to the file **BC_201.n.ib** where *n* is the grid number and *ib* is the BC number. As in the OVERFLOW Q file header, **TVREF** and **DTVREF** are the current simulation time and time step, non-dimensionalized by V_{ref}.

PLOT3D Function File Format

PLOT3D function files are used for reading and writing field variables for several different utility functions. One example is for specifying transition region **ITTYP**=104. PLOT3D function files are Fortran unformatted, and are of the form

```
         READ(1) NJ,NK,NL,NF
         READ(1) F(1:NJ,1:NK,1:NL,1:NF)
```

Here (NJ,NK,NL) are the dimensions of the grid region, and NF is the number of field variables in the array F.

fomoco.out File Format

The following force/moment/mass flow information will be written to the **fomoco.out** text file for each component at each **NFOMO** time step as specified in the **&GLOBAL** NAMELIST:

Component name

ITER	-COMPNO	AREA	AREA_X	AREA_Y	AREA_Z
$C_{Xpressure}$	$C_{Ypressure}$	$C_{Zpressure}$	$C_{Xviscous}$	$C_{Yviscous}$	$C_{Zviscous}$
$C_{Xmomentum}$	$C_{Ymomentum}$	$C_{Zmomentum}$	$C_{Lpressure}$	$C_{Dpressure}$	$C_{Spressure}$
$C_{Lviscous}$	$C_{Dviscous}$	$C_{Sviscous}$	$C_{Lmomentum}$	$C_{Dmomentum}$	$C_{Smomentum}$
C_{Mroll}	C_{Mpitch}	C_{Myaw}	Mass Flow	Time	
$C_{MXpressure}$	$C_{MYpressure}$	$C_{MZpressure}$	$C_{MXviscous}$	$C_{MYviscous}$	$C_{MZviscous}$
$C_{MXmomentum}$	$C_{MYmomentum}$	$C_{MZmomentum}$			

The C_x, C_y, C_z, C_{mx}, C_{my}, and C_{mz} coefficients are written in the global coordinate system. The C_D, C_L, and C_S coefficients are written in the wind axis system (see Chapter 3 for more detail). The Mass Flow and the coefficients with the momentum subscript are calculated when fomoco is applied to non-solid surfaces. Time is the incremental wall clock time.

sixdof.out File Format

The **sixdof.out** file is a text file with the format

```
C       First record.
        READ(1,501) ISTEP,DTVREF,NBODY
C       Read position and orientation at each time step.
        DO NN=1,NSTEPS
        DO NB=1,NBODY
          READ(1,501) NB
          READ(1,502) E1,E2,E3,E4
          READ(1,503) X0,Y0,Z0,UR,VR,WR
          READ(1,503) WX,WY,WZ,WJ,WK,WL
          IF (I6DOF.EQ.1) READ(1,503) FX,FY,FZ,TX,TY,TZ
         ENDDO
         ENDDO
  501   FORMAT(I7,ES15.7,I5)
  502   FORMAT(4ES22.14)
  503   FORMAT(6ES15.7)
```

NBODY is the number of bodies in the simulation. NSTEPS is the number of time steps in the moving body simulation. The variables E1, E2, E3, E4, UR, VR, WR, WX, WY, WZ, WJ, WK, and WL are defined in the **&SIXINP** NAMELIST input (Appendix B). X0, Y0, and Z0 are the location of the body center-of-gravity in the global coordinate system. FX, FY, and FZ are the non-dimensional forces applied to the body in the global axis system. TX, TY, and TZ are the non-dimensional torques applied to the body about the body CG in the global coordinate system. The non-dimensionalization of the forces and moments is shown in Chapter 5.

animate.out File Format

The **animate.out** file is a text file with the format

```
C       Skip first record.
        READ(1,*)
C       Read CG location in body coordinates of each body.
        T00 = -1.
        DO NB=1,NBODY
          READ(1,501) NB,T00,X00,Y00,Z00,E1,E2,E3,E4
        ENDDO
C       Read initial position of each body.
```

```
      DO NB=1,NBODY
        READ(1,501) NB,TN,X0N,Y0N,Z0N,E1N,E2N,E3N,E4N
      ENDDO
C     Read position and orientation at each time step.
      DO NN=1,NSTEPS
      DO NB=1,NBODY
       READ(1,501) NB,TIME,X0,Y0,Z0,E1,E2,E3,E4,
     &             UR,VR,WR,WX,WY,WZ,WJ,WK,WL,
     &             FXP,FYP,FZP,FXV,FYV,FZV,MX,MY,MZ,SA
      ENDDO
      ENDDO
  501 FORMAT(I6,4ES15.7,4ES22.14,19ES15.7)
```

NBODY is the number of bodies in the simulation. NSTEPS is the number of time steps in the moving body simulation. The variables X00, Y00, Z00, E1, E2, E3, E4, UR, VR, WR, WX, WY, WZ, WJ, WK, and WL are defined in the **&SIXINP** NAMELIST input (Appendix B). TN is the non-dimensional initial time at the start of the movement, and TIME is the non-dimensional time at each time step. The time non-dimensionalization is described in Chapter 5. FXP, FYP, and FZP are the pressure forces applied to the body in the global coordinate system, and FXV, FYV, and FZV are the viscous forces applied to the body. MX, MY, and MZ are the moments applied to the body about the body CG in the global coordinate system. The non-dimensionalization of the forces and moments is shown in Chapter 5. SA is the surface area (not the reference area) of the body.

contact.out File Format

The **contact.out** file is a text file written after each contact event is detected between two bodies in a moving body simulation. The file has the format

```
      WRITE(1,1) ISTEP,IDA,IDB,ITER
      WRITE(1,3) V_IMPACT
      WRITE(1,3) PA
      WRITE(1,3)UN
      WRITE(1,3) FIA
      WRITE(1,3) TIA
      WRITE(1,3) FIB
      WRITE(1,3) TIB
      WRITE(1,3) VA
      WRITE(1,3) VB
      WRITE(1,3) WA
      WRITE(1,3) WB
    1 FORMAT(4I10)
    3 FORMAT(10X,3E15.7)
```

ISTEP is the iteration number when the contact occurred. IDA and IDB are the Body ID numbers for the bodies involved in the collision. ITER is the contact iteration number (for debugging information only). PA is the contact point for the bodies (x, y, z). UN is the unit vector for collision forces (x, y, z). FIA is the impact force on body A (F_x, F_y, F_z). TIA is the impact torque on body A (T_x, T_y, T_z). FIB is the impact force on body B (F_x, F_y, F_z). TIB is the impact torque on body B (T_x, T_y, T_z). VA is the post-impact linear velocity for body A (V_x, V_y, V_z). VB is the post-impact linear velocity for body B (V_x, V_y, V_z). WA is the post-impact angular velocity for body A (W_x, W_y, W_z). WB is the post-impact angular velocity for body B (W_x, W_y, W_z).

STOP File

After each iteration, OVERFLOW 2.3 checks for the existence of a file named **STOP** in the current directory. If this file exists, the code will save the current flow solution and stop. This can be used to stop a run gracefully, without losing the work done so far. For example, if a job has been running for an extended length of time, one can go to the directory where the job was started from and type **touch STOP**. This creates a (zero-length) file **STOP**. When OVERFLOW 2.3 finishes the current iteration, it finds the file, saves **q.save**, and stops. As an alternative, a **STOP** file can be created containing an iteration number. If OVERFLOW has passed this iteration, the code will write **q.save** and stop at the end of the current iteration; otherwise it will stop at the end of the specified iteration. The **STOP** file is deleted by OVERFLOW 2.3 before exiting. The **overrun** and **overrunmpi** scripts will check for a **STOP** file before starting. If found, the run will be aborted. (The contents of the file are not checked.)

SAVE File

OVERFLOW 2.3 also checks for a **SAVE** file after each iteration, though this check is only done during iterations on the fine-grid level. If the **SAVE** file contains an iteration number, a **q.save** file is written after that iteration, and the run continues. If no number is present, or the iteration is already past, a **q.save** file is written after the current iteration.